

**AFRL-PR-WP-TP-2007-228**

**ADAPTIVE DISTRIBUTED  
INTELLIGENT CONTROL  
ARCHITECTURE FOR FUTURE  
PROPULSION SYSTEMS  
(PREPRINT)**



**Alireza R. Behbahani**

**APRIL 2007**

**Approved for public release; distribution unlimited.**

**STINFO COPY**

**This is a work of the U.S. Government and is not subject to copyright protection in the United States.**

**PROPULSION DIRECTORATE  
AIR FORCE MATERIEL COMMAND  
AIR FORCE RESEARCH LABORATORY  
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7251**

<b>REPORT DOCUMENTATION PAGE</b>				<i>Form Approved</i> OMB No. 0704-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>					
<b>1. REPORT DATE (DD-MM-YY)</b> April 2007		<b>2. REPORT TYPE</b> Conference Paper Preprint		<b>3. DATES COVERED (From - To)</b>	
<b>4. TITLE AND SUBTITLE</b> ADAPTIVE DISTRIBUTED INTELLIGENT CONTROL ARCHITECTURE FOR FUTURE PROPULSION SYSTEMS (PREPRINT)				<b>5a. CONTRACT NUMBER</b> In-house	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b> 62203F	
<b>6. AUTHOR(S)</b> Alireza R. Behbahani				<b>5d. PROJECT NUMBER</b> 3066	
				<b>5e. TASK NUMBER</b> 03	
				<b>5f. WORK UNIT NUMBER</b> 306603TM	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Structures and Controls Branch (AFRL/PRTS) Turbine Engine Division Propulsion Directorate Air Force Research Laboratory, Air Force Materiel Command Wright-Patterson Air Force Base, OH 45433-7251				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b> AFRL-PR-WP-TP-2007-228	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> Propulsion Directorate Air Force Research Laboratory Air Force Materiel Command Wright-Patterson AFB, OH 45433-7251				<b>10. SPONSORING/MONITORING AGENCY ACRONYM(S)</b> AFRL/PRTS	
				<b>11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)</b> AFRL-PR-WP-TP-2007-228	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> Approved for public release; distribution unlimited.					
<b>13. SUPPLEMENTARY NOTES</b> Conference paper submitted to the Proceedings of the 61st Meeting of the Society for Machinery Failure Prevention Technology. This is a work of the U.S. Government and is not subject to copyright protection in the United States. PAO Case Number: AFRL/WS 07-0924; Date cleared: 01 Jun 2007. This paper contains color.					
<b>14. ABSTRACT</b> Adaptive Distributed Control System (ADCS) architectures have been developing rapidly in all fields to provide for optimal control, prognosis and safety-critical systems. Concepts such as controller decentralization and "smart" sensor/actuators have been studied by both government agencies and industry. Distributed control is potentially an enabling technology for advanced intelligent propulsion system concepts and is one of the few control approaches that is able to provide improved component and control system prognostics, as well as fault-tolerance. ADCS architectures offer the potential of enhanced reliability and the ability to maintain optimal performance when the propulsion system degrades. An ADCS will reduce the impact of hardware and software obsolescence. Control system weight will be reduced by replacing heavy harness assemblies and FADECs, with distributed processing elements interconnected. This paper reviews current activities that may lead to the development of standards for distributed, safety-critical and supportable intelligent propulsion systems of the future.					
<b>15. SUBJECT TERMS</b> Distributed Control, Propulsion; Engine, Adaptive Control, Control System Design, Safety Critical, Centralized Control System					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT:</b> SAR	<b>18. NUMBER OF PAGES</b> 56	<b>19a. NAME OF RESPONSIBLE PERSON (Monitor)</b> Alireza R. Behbahani <b>19b. TELEPHONE NUMBER (Include Area Code)</b> N/A
<b>a. REPORT</b> Unclassified	<b>b. ABSTRACT</b> Unclassified	<b>c. THIS PAGE</b> Unclassified			

# **Adaptive Distributed Intelligent Control Architecture for Future Propulsion Systems**

**Al Behbahani**

Air Force Research Laboratory

**Abstract-** Adaptive Distributed Control System (ADCS) architectures have been developing rapidly in all fields to provide for optimal control, prognosis and safety-critical systems. Concepts such as controller decentralization and “smart” sensor/actuators have been studied by both government agencies and industry. Distributed control is potentially an enabling technology for advanced intelligent propulsion system concepts and is one of the few control approaches that is able to provide improved component and control system prognostics, as well as fault-tolerance. ADCS architectures offer the potential of enhanced reliability and the ability to maintain optimal performance when the propulsion system degrades. An ADCS will reduce the impact of hardware and software obsolescence, thereby minimizing one of the biggest cost drivers of propulsion control systems. Control system weight will be reduced by replacing heavy harness assemblies and Full Authority Digital Electronic Controls (FADECs), with distributed processing elements interconnected through a serial bus. Efficient data flow throughout the propulsion system will provide enhanced visibility into operating components and their health status, while minimizing the update rates of the outer control loop. Distributed control will encourage component reusability and cross-platform portability, while enabling the simplified reconfiguration of the control system. Additional benefits from an ADCS include integration of thermal management capabilities, reduced acquisition, maintenance and sustaining costs of propulsion systems and improved transition and implementation of advanced engine health management. This paper reviews current activities that may lead to the development of standards for distributed, safety-critical and supportable intelligent propulsion systems of the future.

**Key Words:** Distributed Control, Propulsion; Engine, Adaptive Control, Control System Design, Safety Critical, Centralized Control System.

## **Introduction**

A Distributed Control System (DCS) refers to a method of control in which the controller element is not centrally located, but is distributed throughout the system. Each element in the overall system is organized into a hierarchy of sub-systems under the command of one or more controllers. A DCS is a collection of sub-systems, each system with its own specific function, interconnected tightly to carry out an integrated data acquisition and control application.

Distributed Control is different from classical control systems in more ways than just the spatial distribution of control elements. Classical control theory proves [1] to be not applicable in modeling distributed control problems where issues of communication delay, jitter, and time synchronization between sub-systems are not trivial. A DCS integrates the subsystems and process controllers of a process line into a coordinated, interactive system. The process is manageable as a complete system, with control over the interrelationship of the various subsystems. A DCS lets you see the “big picture” and improve the overall efficiency and quality of the entire propulsion system. As propulsion systems become larger and more complex, the justification for a DCS grows.

Industry is full of successful examples of DCS application [2- 12]. In the past few years, DCSs have developed very quickly and have been widely applied in numerous fields [2- 12]. In a DCS, controllers and even the components commanded by them are spatially separated from each other, often at great distances. They exchange data amongst themselves in real, or near real-time, depending on the application. The entire system is networked for communication and monitoring, typically using common digital communication standards. Still largely untapped are biological examples of distributed control where autonomous individual organisms perform coordinated tasks.

In most engine control applications, there is an inherent multi-objective optimization problem which is of growing interest in the controls community. When evaluating aero propulsion systems, it is often found that the most efficient operating point of the system does not necessarily match the optimum operating point of the individual components. Distributed Control Systems may be ideally suited to address such issues because intelligence is embedded in components while overall control is maintained in the FADEC.

The need for Distributed Control Systems in gas turbine applications is driven by many factors; among them is cost, weight, and the need for additional information on system operation. In the past ten years, embedding intelligent control directly into sensors and actuators has dramatically increased. In other fields, this has led to an explosion of smart sensors and actuators available from different manufacturers, especially in the process control, manufacturing, chemical and nuclear industries. Moreover, increased applications in aerospace are being found, especially in prognostics and health management. Currently in the aero community, integration of intelligent components is being carried out in an ad hoc manner by incorporating smart elements into inherently centralized architectures. The goal has generally been to reduce harness weight and improve fault detection and isolation. Furthermore, recent advances in wireless technologies, have established the possibility of very short range intelligent communication applications (for health monitoring) within the aircraft.

With the recent advancements in commercial availability of processing power, implementing Distributed Control Systems has become more practical. Advancements in microprocessor and parallel computing, together with real-time systems for prognosis and diagnosis, are key elements in enabling distributing control. As systems become more complex, the ability to employ Digital Signal Processors (DSP), in-process measurement

and control, and parallel computation are powerful tools which can be exploited. For advanced engine control, this may be an excellent opportunity for producing open and distributed fault-tolerant control and monitoring systems.

Distributed Control System technology is complimentary to other technologies being developed for gas turbine engines. A DCS is a toolset that's full value has not been fully quantified. In developing systems for fault detection, isolation and accommodation, and health monitoring, DCSs may offer advantages over a centralized system. Distributed, embedded systems that consist of multiple interacting subsystems with tightly integrated software components can provide fault detection and isolation (FDI) for the safe operation of propulsion systems. Distributed embedded systems can be used to design systematic, scalable, robust systems that are fault tolerant. They can also be used for managing the complexity of the FDI task, as well as being the enabler for model-based FDI algorithms, which exhibit more reliable and robust attributes than centralized systems. These qualities lead to the belief that a DCS is well suited for safety-critical systems.

There are major technical challenges to the realization of full distributed control architecture. With increased demands on flight systems, there is a corresponding need for control components to work reliably in harsh environments and at higher temperatures. The high temperature actuator control module is a critical component for implementing a distributed engine control system. The importance of this technology is demonstrated by its inclusion in DoD programs such as the Integrated High Performance Turbine Engine Technology (IHPTET) program and, more currently, in the Versatile Affordable Advanced Turbine Engines (VAATE) program, both of which specify objectives for reducing weight, production and maintenance costs.

In summary, the objective of this paper is to present and review the need for a DCS in propulsion systems.

### **Adaptive Nature of Distributed Control Capability for an Intelligent Control Design**

To fully realize the potential of intelligent control methodologies, future sensor and actuator technologies must incorporate adaptive control algorithms. An adaptive optimization approach that supervises, and is synergistic with the main controller using intelligent components, is the key to a fault tolerant engine controller. The distributed control concept has an inherent adaptive nature that can be easily incorporated within the distributed controls. An Adaptive Distributed Control System (ADCS) is similar to a Distributed Control System (DCS) and is one of the central enabling technologies needed for an advanced turbine engine to achieve optimum performance. Adaptive distributed control incorporates the capability to intelligently reconfigure itself in the face of an ever-changing set of conditions within the plant and the controller system itself.

The concept of adaptive algorithms can be applied across control topologies, including the DCS architecture. The capability can be repackaged and easily embedded into subsystems. Also known as model-based control or self-identifying systems, these

subsystems can adjust automatically to deviations and can self-optimize to move within a preset tolerance to account for aging, or other condition changes, even as the engine degrades due to wear and tear. The impacts of these deviations are minimized by the controller, thereby achieving optimum engine performance throughout the engine life. Examples of elements of adaptive distributed control include reduced design margins through micro adaptive flow control in the inlet, flow passages, and on airfoils; efficiency improvements through active clearance control; reduced pattern factor, lower emissions, and elimination of combustion instabilities through active combustor control. Aging sensors and actuators which lose sensitivity can incorporate intelligence for self-adjustment. The challenges of integration of active component control and diagnostics technologies into the control of the overall engine system provide additional rationale for moving from the current centralized control systems to distributed control architectures.

### **DCS Consideration for Future Propulsion**

Aircraft engines could be used, operated and maintained more efficiently if we knew more about what was really happening inside them. When considering a DCS for propulsion systems, especially for gas turbine engines, one must systematically evaluate the implementation penalties versus the benefits. From a general perspective, the major benefit of a DCS is improving fault detection and isolation compared with a centralized system. The other major benefit of DCS is flexibility to upgrade and add new components. Today's aircraft and engine systems have to be inspected at regular intervals for evidence of internal damage such as cracking or erosion. This is usually performed using boroscopes or other NDE processes, which are time consuming and expensive. One possible solution is speeding up and automating the inspection process; however, a more attractive solution would be the development of an intelligent engine, which would do this automatically, while monitoring turbine health status in flight.

### **Historical Perspective on Distributed Control and High Temperature Components**

In the 1990's, AFRL started to build the foundations of distributed control. The US Air Force was pursuing aggressive performance and cost goals for future jet engines as part of Department of Defense and NASA Integrated High Performance Turbine Engine Technology (IHPTET) program. Specific goals Studies have shown that better regulation of engine thrust could result in a doubling of the thrust/weight ratio and a 40% fuel burn reduction, while at the same time, providing a 35% decrease in engine life cycle costs [13, 14]. To achieve these metrics, control logic must be designed to use a broad array of hundreds of sensor and actuators operating over a wide operating temperature range, to control thrust while protecting against aerodynamic, thermal, or material strength limitations. All the IHPTET goals may not necessarily be achievable using DCS.

The external casing of an engine is a significant source of radiant heat, in some instances reaching over 450°C at high mach and altitude operations. It is anticipated that a significant number of smart sensing and actuation applications could be added on the

casing, mounting in such a way as to limit the heat soak. This worst-case thermal environment was used as a guide to motivate the development and application of high temperature electronics operating at junction temperatures greater than 250°C.

The High Temperature Distributed Control System (HiTeC) consortium was established to address the above issues. The HiTeC Program, a dual-use (military and commercial) technology development agreement awarded under the 1995 Technology Reinvestment Project solicitation, was sponsored by the Defense Advanced Research Projects Agency (DARPA). United Technologies Corporation, acting through the United Technologies Research Center (UTRC), organized a 14-member consortium of leading aerospace companies and the University of Maryland to further the development of high temperature electronics. Other consortium members include Pratt & Whitney, Boeing Defense and Space Group, Rockwell Science Center, and various actuator and electronics suppliers.

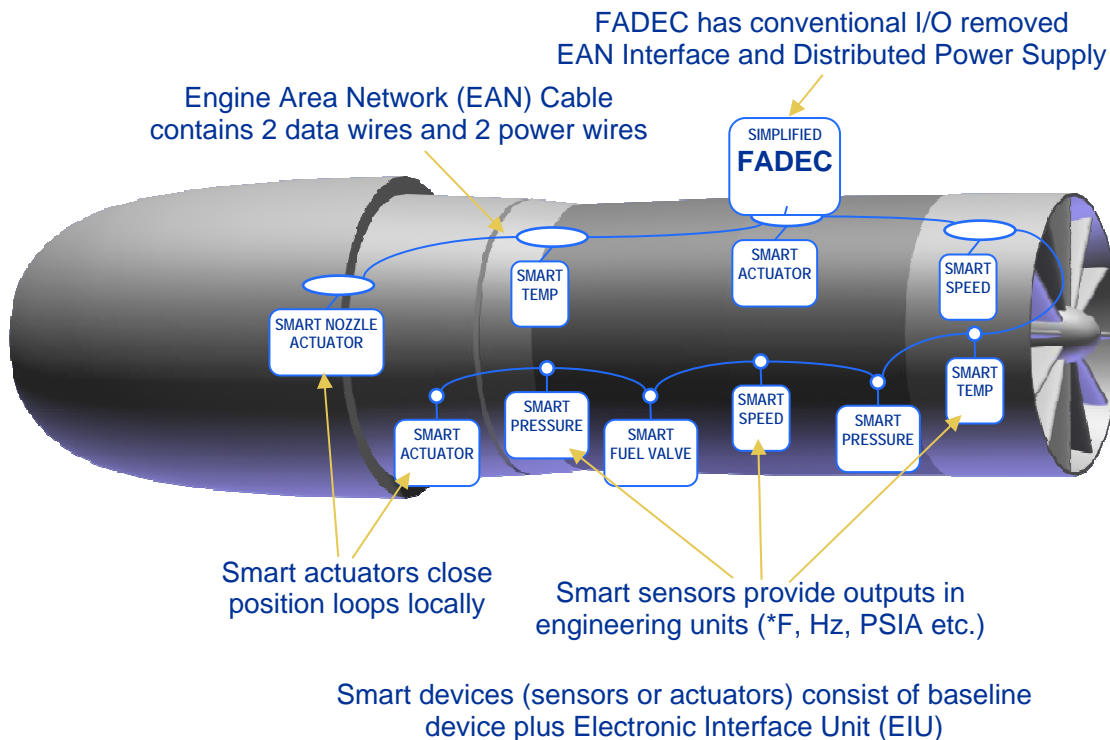
The high temperature actuator control module was identified as the critical component for a distributed engine control system, which was a key to achieving IHPTET Phase III Controls and Accessories' objectives for reducing weight and production and maintenance costs. The demonstration platform for the HiTeC smart actuator was the Pratt & Whitney XTE66 engine. The consortium completed rig environmental testing and a passive engine demonstration of a variable vane actuator control module, which can operate without active cooling up to 225°C (437°F).

The HiTeC consortium was also involved in the development of sophisticated distributed architectures that could improve performance, power output, or fuel economy if the electronics were able to be located close to the control target, as seen in the FADEC system shown in Figure 1.

This distributed architecture is a more effective means of controlling the engine due primarily to an extreme reduction in harness complexity (reduction in wire count from >500 to as few as 8) and improved ability to isolate failures in the system. The FADEC is freed of tedious loop closure and signal conditioning tasks, performing only high level control law computations. The resulting simplification of the interface between the FADEC and the rest of the control system enables its integration into the aircraft avionics suite, with an accompanying cost reduction (from elimination of the unique cooling loop and environmental hardening aspects of its design) that offsets the cost increase at the distributed modules.

Although there have been other similar initiatives by the government and industry, these ideas have not been transitioned to the Original Engine Manufacturers (OEMs) in the propulsion field. Pratt & Whitney worked on a Distributed Control System Architecture using common distributed control technologies in the 1990's. Various architectural configurations, labeled "Partially Distributed", "Minimally Distributed", and "Fully Distributed", were also developed. P&W also prototyped a Common FADEC for military and commercial engine products and funded other programs such as the Optoelectronic FADEC. General Electric, together with BAE Systems, developed the

idea of the Flexible FADEC. The Modular Aerospace Controller (MAC) was developed by Honeywell and was transitioned on to the F124 and F110 engines; other applications are pending. In 2003, AFRL investigated the Universal FADEC which included aspects of distributed engine control for both hardware and software. For a limited time a consortium was formed to work on a common platform. The Commercial Off-The-Shelf (COTS) FADEC concept, by P&W, was perhaps one of the most innovative ideas using modular design components from P&W commercial engines. Although modular, it was not based on open standards. Goodrich also extended the idea of a Universal FADEC for helicopter applications. There have been numerous other activities; however, none have been adopted for production by major OEMs due to high development cost.



**Figure1: High Temperature Distributed Control Systems “HiTeC”**

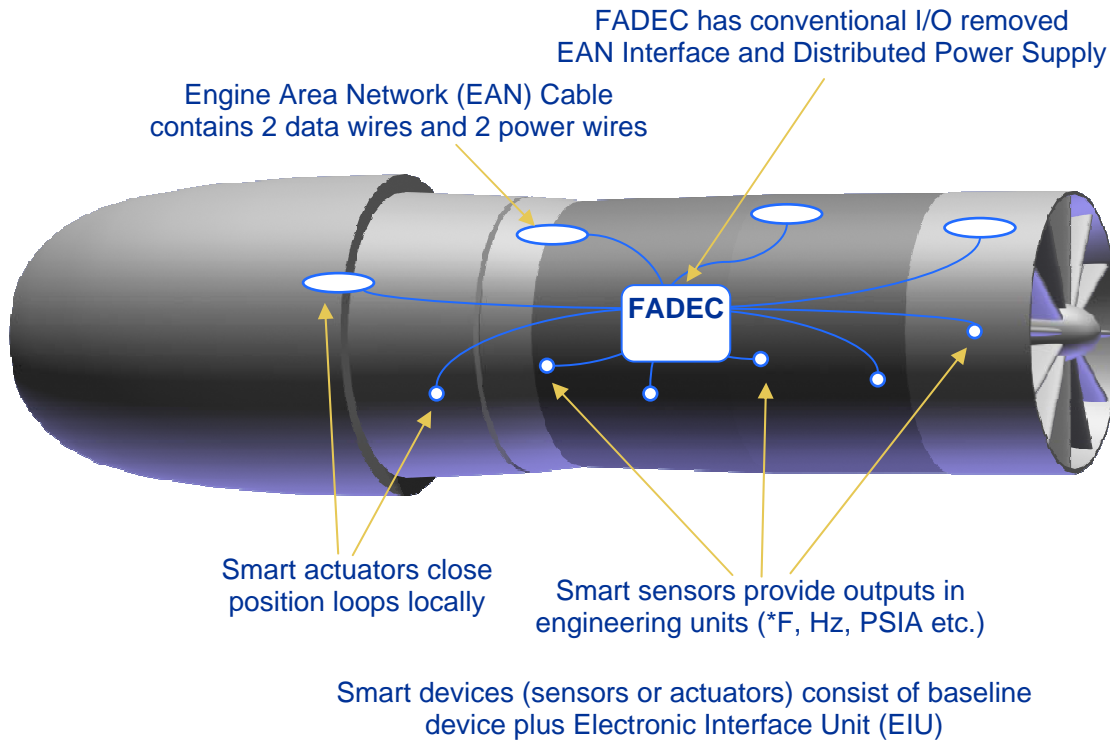
### Distributed Versus Centralized control System

In order to evaluate distributed versus centralized control system schemes or algorithms, one must understand the advantages and disadvantages for each.

In the traditional Centralized Control System (CCS) configuration, shown in Figure 2, the centralized control processor handles all processing functions, including; the operating system, task scheduling, I/O, protection, communication, and control algorithms. In the CCS scheme or platform, we assume that there is a centralized controller such as FADEC, which manages both application software (AS) as well as operating system (OS) architecture for control logic, scheduling and I/O connections. All computations are performed by a single controller and the control signals are transmitted to each individual



datum location in the array; that is, sensor and actuator information are shared globally. In such a scheme, we always assume to have correct information about the network of control elements and all information is current. The computing capacity of the centralized FADEC, therefore, has to be significantly higher than in a DCS. However, there is only one CPU with expansive storage capacity - which means only one such spare part is needed. The CCS processor has to poll the network every X seconds, so, although it will know the network status, how current the information actually is, depends on the frequency of probing the network.



**Figure 2: Current Centralized control System for Turbine Engine.**

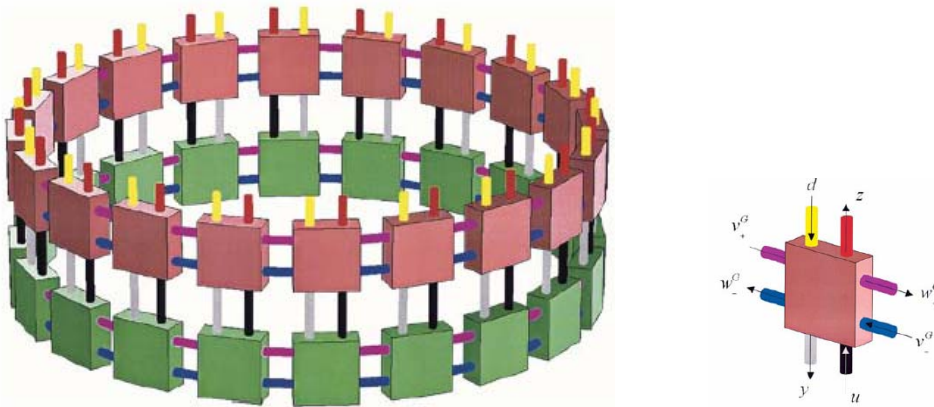
Alternatively, DCS schemes call for distribution of the computational capability across a network of smaller systems. The level of distribution can be considered a continuum and is subject to the choices of the designer. There are many strategies that can be employed to control spatial array systems, including centralized, fully de-centralized, or distributed [2]. In one distributed control scenario, specific sensors and actuators are connected only to the local controller, which, operating independently does not need to share its low level information globally. However, there may be dynamic interactions between this network of neighboring modules, making the overall system interactions more complex.

There are many distributed control schemes. Distributed control design for Spatially Interconnected Systems was studied in detail by Raffaello D'Andrea and Geir E. Dullerud [15]. A major advantage of the DCS architecture includes the simplification of cable harnesses and the reduction of wiring weight. The functional compartmentalization

of DCS architecture is also convenient for adding a new component or modifying the control logic of an existing function, without drastically changing the design, thereby minimizing obsolescence. There are also disadvantages of a DCS. These disadvantages include the necessity of embedding processing capability in multiple locations and the environmental and integration issues associated with each module.

The application of distributed control technology will have the added benefit of facilitating fault isolation with 100 percent certainty, not only improving in-service reliability, but also providing savings in the cost of maintenance of that system over a typical engine controller. Another advantage of such a DCS design approach is scalability. We can add additional modules to the same system without having to modify the lower level control algorithms; the limiting factor being communication bandwidth and latency. The FADEC will also scale by adding processing power and memory, but since much of the processing is accomplished remotely, it will not scale as quickly as in a CCS. The control of such systems is typically performed locally in a distributed manner to ensure a simple, scalable, yet robust system. However, such distributed control systems are less efficient and more restricted than a centrally served system. In many situations, a control system with a global view is required where a local view can not provide sufficient information for operation [16].

One factor which may discourage the widespread adoption of a DCS, is the lack of global performance and stability [2]; however, techniques have been developed for application of DCSs in spatial array systems [15], [17]. This type of control implementation is neither centralized nor completely de-centralized (Fig.3).



Basic building block for control design, one spatial dimension.

**Figure 3: Spatial Array systems - Interaction between the controllers using closed loop system, periodic interconnection, and one spatial dimension.**

In this scheme, each individual in the array is equipped with a controller which has a particular spatial structure. The structure determines the extent to which local information is shared between the neighboring units. We can consider the application of

such techniques to distributed control system for turbine engine control. Table 1 presents a comprehensive comparison of CCS and DCS characteristics.

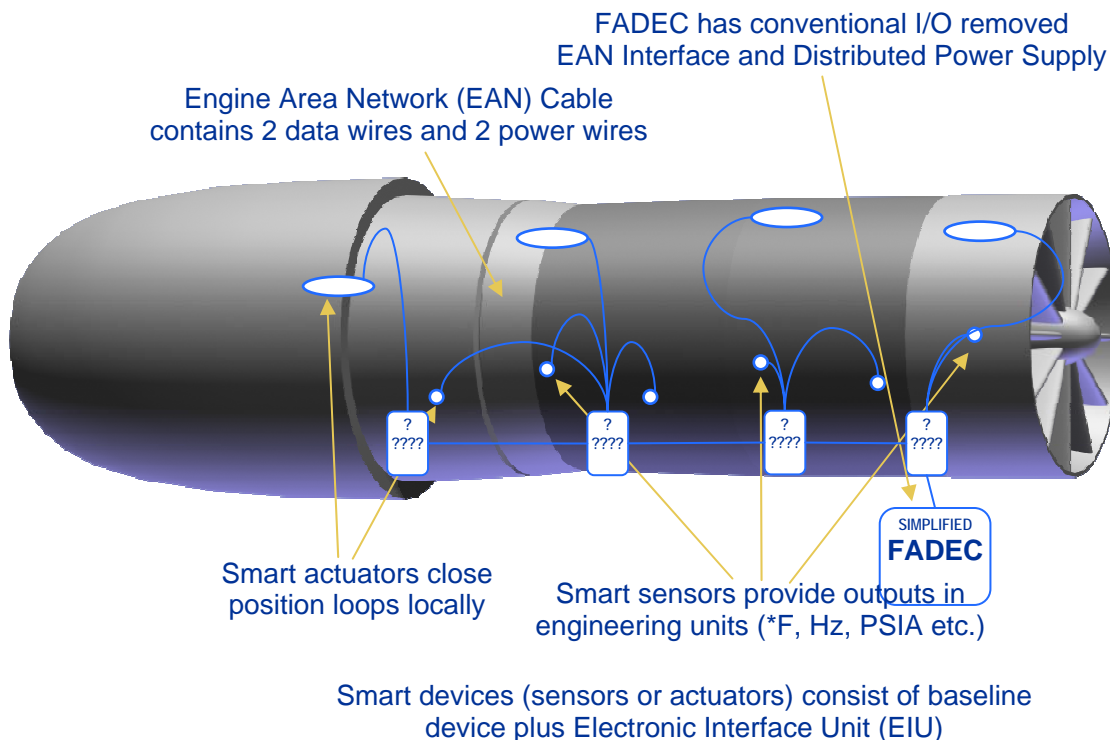
**Table 1: Characteristic comparison of CCS and DCS.**

<b>Hardware / Software Feature Comparison:</b>	<b>CCS</b>	<b>DCS</b>
<i>controller or manager or processor</i>	centralized	distributed
<i>number of controllers or processors</i>	1	>1
<i>impact of obsolescence</i>	major	minimal
<i>status of links / network failures at any time</i>	✓	
<i>fault isolation with 100% certainty</i>	no	yes
<i>in-service reliability</i>		high
<i>cost of maintenance</i>	high	low
<i>harness weight / complexity</i>	high	low
<i>ease of adding system functionality</i>	hard	easy
<i>Ease of centrally located repair of one component</i>	✓	
<i>Device location plan and design</i>	✓	
<i>Accuracy of status of each component</i>		✓
<i>Impact of processor failure on overall system</i>	major	
<i>Ease of design of control algorithms</i>		✓
<i>quantity of CPUs</i>	2 with redundancy	large
<i>Each processor communicate with the master CPU</i>		✓
<i>Computing capacity of Central CPU</i>	very high	moderate
<i>reusability and maintenance of the software</i>	✓	✓✓✓
<i>Synchronization</i>	✓	✓
<i>Ease of Replacement and testing of individual modules</i>		✓
<i>Capabilities and user interfaces</i>		
<i>Scalability.</i>		high

### **Decision on distributed fault-tolerant control and monitoring systems**

DCS local control technology offers a very structured architecture, and replacement and testing of individual modules is straightforward. Due to the simple topology,

standardized buses can be used without problems. Communication among control modules, with the master FADEC, and for synchronizing the controllers during start-up and stopping, has to be programmed to minimize instabilities. Central diagnostics, commissioning and maintenance are the principal arguments for using central control technology, as are start-up and stopping of the plant, as well as administration of the single controller. A distributed architecture can include some or all of these attributes, depending on the tools and design. If the central FADEC is powerful enough, it can be used to control and synchronize a large number of modules for any expansion. Other criteria for selecting which architecture to use include flexibility and reusability, and costs in terms of the hardware, wiring, commissioning and configuration. Each individual unit in the array is equipped with a controller which has a particular spatial structure. The structure determines the extent to which local information is shared between the neighboring units. For FADEC applications, a DCS can resolve thermal issues without significantly adding weight or requiring new technologies. Recent work has indicated that stability of the control system may be an issue, however, research has shown that sharing information in multi-directions (e.g. a "lookahead" scheme, as well as "lookback"), will improve stability [18]. Figure 4 shows a version of a distributed engine control application for gas turbine engines. In this scheme, which is a version of distributed control, there are several nodes or sub-controllers that are inter-connected by communication lines which are described in the next paragraph.



**Figure 4: A Distributed Control Systems for engine Control**

An indepth study to investigate the feasibility of a DCS application for propulsion systems would be beneficial. A DCS application with modular integrated, distributed components is envisioned. Communication of information from one distributed component to another is an essential service provided by modular architectures that support the functions of a DCS. The communication bus structure and the protocols used are the principal mechanisms to consider. There are many existing and developing Open System communication standards, among them are; SAFEbus, TTTech Time-Triggered Architecture (TTA) , FlexRay, CAN, ARINC 429, RS232, IEEE488, NASA SPIDER, MIL-STD-1553, IEEE 1394b, SpaceWire, Ethernet 10/100 Base-T, Avionics Full-Duplex Switched Ethernet, Fibre Channel , Gigabit Ethernet, etc.

Several studies regarding buses to communicate between modules have been completed [19]. NASA compared bus architectures for safety-critical embedded systems [20]. The modular architectures that support avionics and control systems for aircraft use distributed fault-tolerant computer systems to provide safety-critical functions such as flight and engine control. They must provide mechanisms for coordinating the distributed components that support a single function (e.g., distributing sensor readings and actuator commands). Additional mechanisms must be incorporated to replicate data to perform the function in a fault-tolerant manner, while protecting functions from faults in each other. Such architectures must tolerate hardware faults in its own components and must provide very strong guarantees on the correctness and reliability of its own mechanisms and services. In the NASA studies, the architecture of two avionic and two automotive buses were compared and their similarities, differences and design trade-offs were examined. The avionics buses considered were the Honeywell SAFEbus (the backplane data bus used in the Boeing 777 Airplane Information Management System) and the NASA SPIDER (an architecture being developed as a demonstrator for certification under the new DO-254 guidelines). The automobile buses considered were the TTTech Time-Triggered Architecture (TTA), recently adopted by Audi for automobile applications, and by Honeywell for avionics and aircraft control functions, and FlexRay, which is being developed by a consortium of BMW, DaimlerChrysler, Motorola, and Philips.

Based on these studies, TTA may be an attractive bus option. TTech provided the benefits of using Time Trigger Pulse (TTP) in Distributed Real-Time Systems [21]. These benefits are presented:

1. Precise Interface Specifications
2. Composability
3. Reusability of Components
4. Improved Relationship between Supplier and Sub-supplier
5. Timeliness
6. Error Containment
7. Constructive Testability
8. Seamless Integration of Fault Tolerance
9. Simpler Application Software
10. Shorter Time-to-Market
11. Reduced Development Costs

## 12. Reduced Maintenance Costs

The TTP technologies, and benefits as an enabling tool for a communication protocol for design of composable distributed real-time systems which operates as dependable as time, should be further examined. Using NASA reports as baseline [19, 20] Honeywell and Hamilton Sundstrand incorporated TTP technologies in their applications. Honeywell used TTP technology for their MAC FADEC, whilst Hamilton uses TTP technologies in the Boeing 787 Dreamliner.

One aspect of distributed controls involves embedding intelligence in components such as sensors and actuators. The smart sensors and actuators, and other key components of distributed controls, need to be explored extensively by the OEMs. It has been recommended that further research is needed to develop appropriate modeling tools that facilitate the design and optimization of large scale distributed systems whose interactions, to a large extent, probabilistic [22].

### **Conclusion**

Efforts have been expended to understand the implications of distributed control for propulsion engine applications. The technology has been well established in other industries. For gas turbine engines, harsh operating conditions and the severe consequences of failure have prevented the use of DCSs. There are two main issues which must be addressed for application of distributed control architectures to gas turbine engines. First, there is a lack of a suitable digital interface standard for connecting the smart sensors and actuators to the simplified FADEC. Secondly, the use of smart sensors and actuators requires that more processing capability be moved to these components which, in turn, require electronics with higher temperature capabilities. There is progress being made in both these areas, and the new government/industry consortium, Distributed Engine Control Working Group (DECWG), is addressing these issues.

The need for the integration of low cost, open, high performance, dependable, distributed fault tolerant control systems in reduced timescales, is required for successful DCSs. A major inhibiting factor to the adoption of Open DCSs, is that control related components are proprietary to the OEM and FADEC manufacturers. This makes integration of systems using COTS components from different manufacturers difficult. There is a need for a toolset that will address the management of requirements, as well as the management of complexity, performance assessment (via multidisciplinary co-simulation), prior to implementation, and management of component obsolescence for applications with a long lifetime.

The payoff of successful implementation of Distributed Engine Control (DEC) could be an increase in engine performance due to the addition of new technologies. The use of local nested control loops as was shown in Figure 4 could relieve the burden on the central FADEC, allowing higher response and tighter control of local processes. There could be a significant reduction in costs in all phases of engine development and

operational deployment. Engine development time could be reduced due to the standardized control element interfaces, and proven and predictable component performance. Engine control component costs could also be reduced, due to increased competition among component suppliers. Operational costs could be reduced due to the reduced impact of component obsolescence. Control components with standard interfaces will isolate the larger control system from any and all internal issues, because they perform as functional elements. Standard interfaces will also enable cross-platform compatibility of components and lower logistic and training costs for servicing engines. The increased awareness of the system performance over time, will positively impact the maintenance costs.

### **Acknowledgment**

The author would like to thank Dennis Culley of the Advanced Controls Technology Branch at NASA Lewis Research Center and Dr. Charles Cross, Structures and Controls Branch Chief from Air Force Research Laboratory for their suggestions and comments regarding this paper.

### **References**

[1] Martin Torngren, Jad Elkhoury, Martin Sanfridson and Ola Redell. *Modelling and Simulation of Embedded Computer Control Systems: Problem Formulation*, Technical Report Mechatronics Lab, Department of Machine Design, Royal Institute of Technology, KTH, S-100 44 Stockholm Sweden, Stockholm, 2001.

[2] Puneet Sharma and Carolyn Beck, *Modeling and Distributed Control of Mobile Offshore Bases*, Proceeding of the 2004 American Control Conference, Boston, Massachusetts, June 30-July 2, 2004. Page 5238 – 5243.

[3] Stubbs and Dullerud, *Networked control of Distributed Systems: a Testbed*, ASME 2001 International Mechanical Engineering Congress and Exposition. 2001.

[4] Stubbs, Vladimerou, Vaughn and Dullerud, *Design of a Vehicle Network Control Testbed*, ACC 2002.

[5] Terry L. Huntsberger, Ashitey Trebi-Ollennu, Hrand Aghazarian and Paul S. Schenker, Paolo Pirjanian, Hari Das Nayar, *Distributed Control of Multi-Robot Systems Engaged in Tightly Coupled Tasks*, Autonomous Robots 17, 79–92, Kluwer Academic Publishers. (The Netherlands), 2004.

[6] Katrin Klinger, *Successful Development of High-Confidence Architecture for Distributed Control Applications*, TTTech, <http://www.vmars.tuwien.ac.at/projects/nexttta>, <http://www.tttech.com/press/pressreleases.htm>, Vienna, Austria, June 29, 2004.

- [7] Yan, G. ; L`Archeveque, J.V.R., *Distributed control and instrumentation systems for future nuclear power plants*, IEEE Trans. Nucl. Sci., V. NS-23, No. 1, pp. 431-435; nuclear science symposium and nuclear power systems symposium; 19 Nov 1975; San Francisco, CA. 1976 Feb 01
- [8] Harold W. Nelson, *Distributed Control System Network Architecture* - WESTREG 2002 Instrument Society of America (ISA) publications, TP02WES025, 2002.
- [9] Murty V. Rama, *Distributed Control System as an Alternative Enterprise Resource Planning Tool*, Instrument Society of America (ISA) publications, TP99UN003, INTERKAMA, 1999.
- [10] Jean B. Hugues (SEMY Engineering, Inc.), Sam L. Sundaram (Motorola, Inc.), *Distributed Control Systems In Automated Ic Manufacturing*, IEEUCHMT '90 JEMT Symposium, pp155-161, 1990.
- [11] Szer-Ming Lee, Robert Hamison and Andrew A. West, *A Component-based Distributed Control System for Assembly Automation*, IEEE, 2004.
- [12] Federico Cuedea, Insop Song and Fakhreddine Kmay, *Distributed Control Framework Design and Implementation for Multi-Robotic Systems: A Case Study on Block Manipulation*, Proceedings of the 2004 IEEE, International Symposium on Intelligent Control, Taipei, Taiwan. September 2-4, 2004.
- [13] T. Lewis, AFRL/PRTA, HiTeC: *Smart Actuator Gets Passing Grade*, USAF Accomplishment Report, Propulsion Directorate, May 1999.
- [14] *High Temperature Distributed Control System (HiTeC) Final Report*, sponsored by the Defense Advanced Research Projects Agency (DARPA) and directed by AFRL's Propulsion Directorate, 1995-1999.
- [15] Raffaello D' Andrea and Geir E. Dullerud, *Distributed Control Design for Spatially Interconnected Systems*, IEEE Transaction on Automatic Control, Vol. 48, No. 9, September 2003, page 1478-1495, 2003.
- [16] L. E. Parker, *Local versus global control laws for cooperative, agent teams*. MIT AI memo no. 1357, March 1992.
- [17] B. Bamieh, F. Paganini, and M. Dahleh, *Distributed Control of Spatially Invariant Systems*, IEEE Transaction on Automatic Control, 1998.
- [18] P. J. Seiler. *Coordinated Control of Unmanned Air Vehicles*. PhD thesis, University of California, Berkeley, 2001.
- [19] D.A. Gwaltney and J.M. Briscoe, *Comparison of Communication Architectures for Spacecraft Modular Avionics System*, Marshall Space Flight Center, Marshall Space Flight Center, Alabama, NASA Report, NASA/TM-2006-214431.



[20] John Rushby, SRI International, Menlo Park, California, *A Comparison of Bus Architectures for Safety-Critical Embedded Systems*, NASA/CR-2003-212161, National Aeronautics and Space Administration, Langley Research Center Hampton, Virginia, 23681-2199 March 2003.

[21] TTTech technology, Time Triggered Technology, website:  
<http://www.tttech.com/index.htm>.

[22] Nikolaus Correll, Christopher Ciani, Xavier Raemy and Alcherio Martinoli, *Self-Organized Embedded Sensor/Actuator Networks for "Smart" Turbines*, IEEE.

[23] R. D. Andrea. *A Linear Matrix Inequality Approach To Decentralized Control Of Distributed Parameter Systems*. Pmc. Of American Control Conference, pages 135G1354, 1998.

[24] B. A. Francis, J. C. Doyle, and A. R. Tannenbaum, *Feedback Control Theory*, McMillan Publishing Company, 1992.

[25] D. E Chichka, I. D. Wolfe, and J. L. Spayer, *Decentralized controllers for unmanned aerial vehicle formation flight*. American Institute of Aeronautics and Astronautics, 1996.

[26] *Vehicle following control design for automated highway systems*. IEEE Control System, 16(6):43-60, 1996.

[27] P. J. Seiler. *Coordinated Control of Unmanned Air Vehicles*, PhD thesis, University of California, Berkeley, 2001.



# MFPT



## Adaptive Distributed Intelligent Control Architecture for Future Propulsion Systems

**Presented at 61st MEETING of the SOCIETY FOR MACHINERY FAILURE  
PREVENTION TECHNOLOGY**

Ramada Plaza Resort Oceanfront Hotel in Virginia Beach, Virginia

April 17-19, 2007

**Al Behbahani, Ph.D.**

**Controls and Engine Health Management**

**Structures and Controls Branch**

**Paper: Session 5B - Cape Colony B**  
Cleared for Public Release, AFRL-WS 07-0924



# AFRL

**THE AIR FORCE RESEARCH LABORATORY**  
LEAD | DISCOVER | DEVELOP | DELIVER



# Presentation Overview

## Distributed Control Systems (DCSs)



- Motivation
- Control System
- Control Architecture
- Different Schemes of Control System Premise
- Comparison of DCS vs. CCS
- Issues to consider
- Summary



# Motivation for the Distributed Control

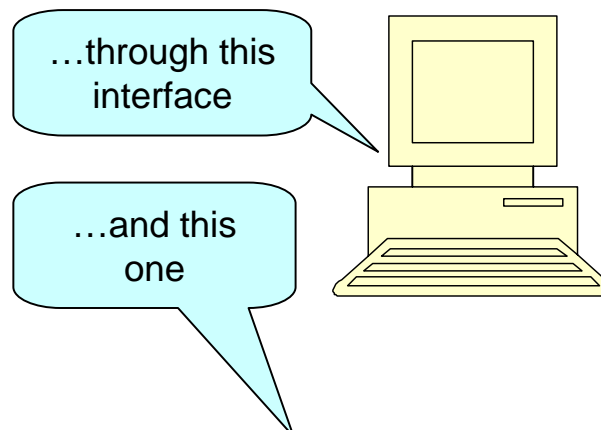
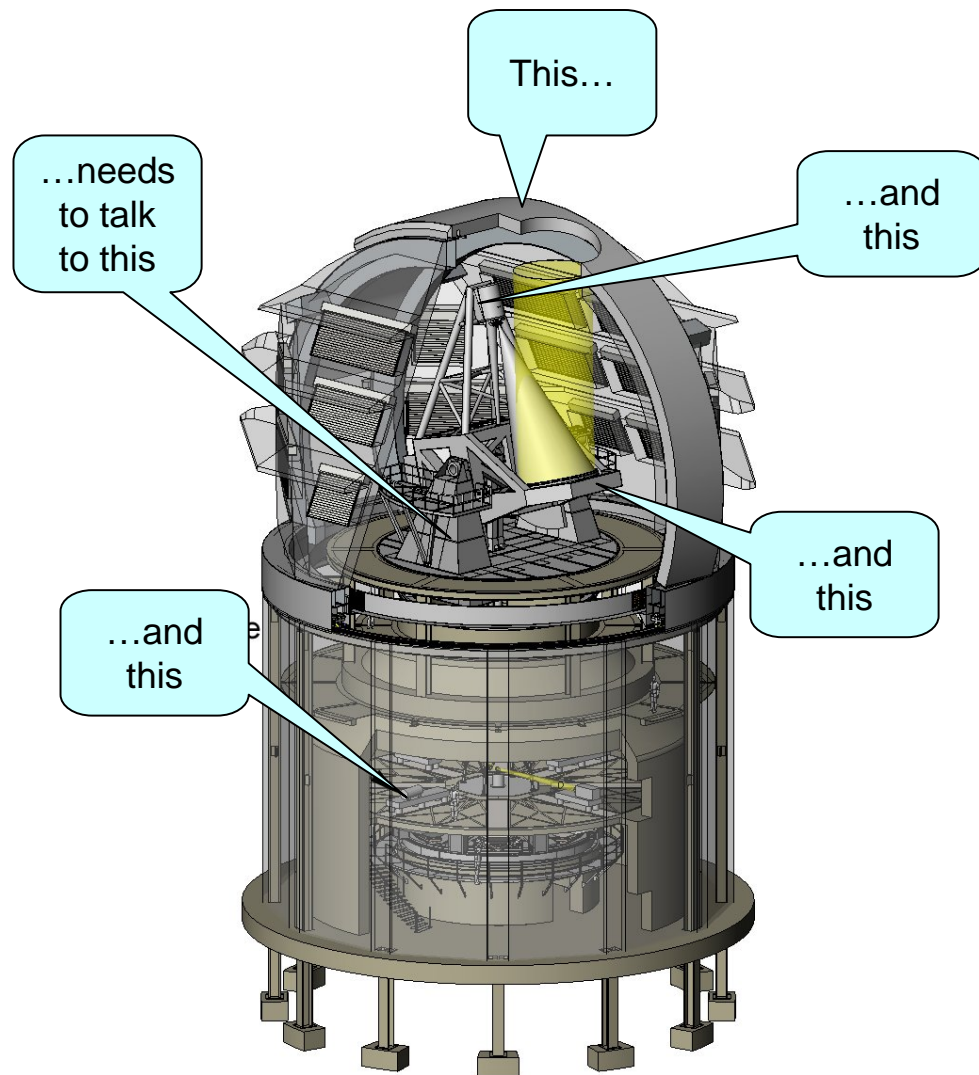


## Articulate the challenges and opportunities for Propulsion Control

- Present a vision that can be used to inform high level decision makers and OEM of the importance of Control design that will affect the entire propulsion system
- Identify possible Challenges and new opportunities
- Provide a compelling view of the field that continues to attract the brightest scientists, engineers, and mathematicians to the field
- Respond to the changing nature of control, dynamics, and systems research



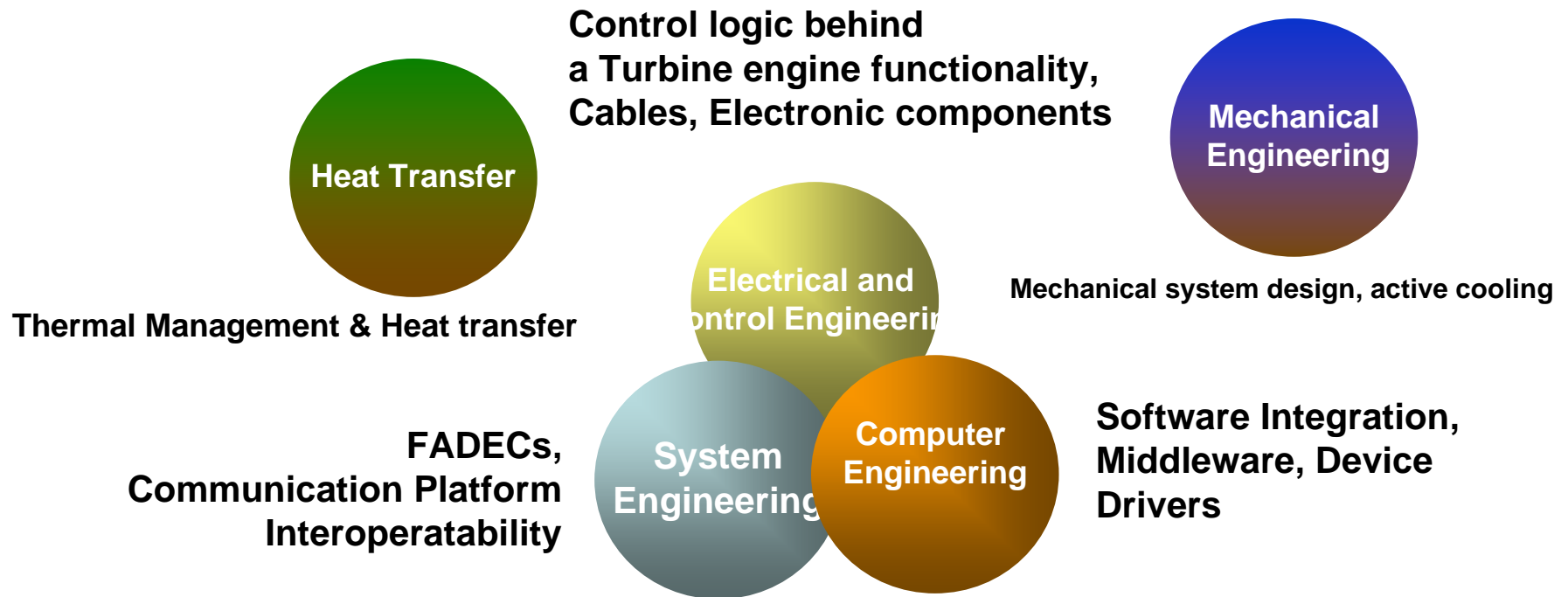
# What is Propulsion Control?



AFRL/ PRTS



# Control System Challenges for Propulsion System



**FADEC Controller design involves  
Many steps requiring many backgrounds**



## Develop Control Decision *Logical* Architecture



- Functional requirements for the control system are defined by:
  - ✓ Definitions and attributes of control decisions
  - ✓ Control decision logical architecture
- Consider providing redundant logic for the same function to enhance reliability and survivability
  - ✓ Recognize trade-off of increased complexity and cost

**Control decision logical architecture provides a basis for the synthesis of candidate hardware architectures that meet the same functional requirements.**



# Propulsion System - Decision *Logical* Architecture



- ***Propulsion Level*** Control Decisions (Flight Control)
  - ✓ Provide information to the pilot to isolate failures and maintain aircraft safety
  - ✓ Provide information to the pilot about the operational status of aircraft components
  - ✓ Do other alternatives if operating component (s) is failing
- ***Engine Level*** Control Decisions (FADEC)
  - ✓ Isolate failures by closing valves closest to the failure or moving actuators
    - Redundant with device level failure isolation
    - Flow balance logic
      - Inputs required from multiple valves
      - Logic is different from engine level to minimize common mode failure
  - ✓ Assess material condition and readiness of an engine
    - Inputs required from multiple devices
- ***Device Level*** Control Decisions (smart sensors, actuator, ...)
  - Isolate failures by closing valves, or other actions closest to the failure component
    - Failure path logic
      - Inputs and actuator are at the valve

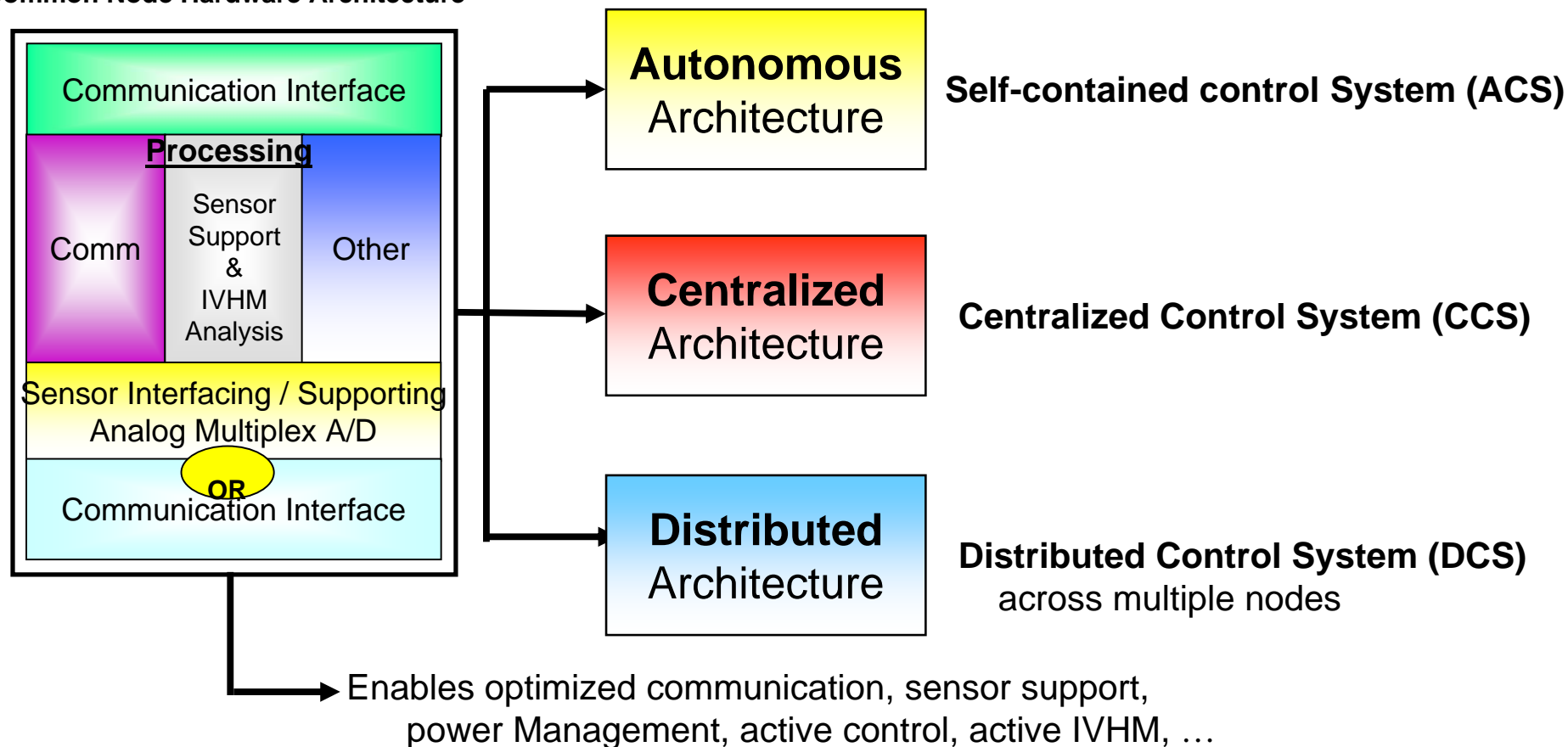




# Different Schemes of Control System



## Common Node Hardware Architecture

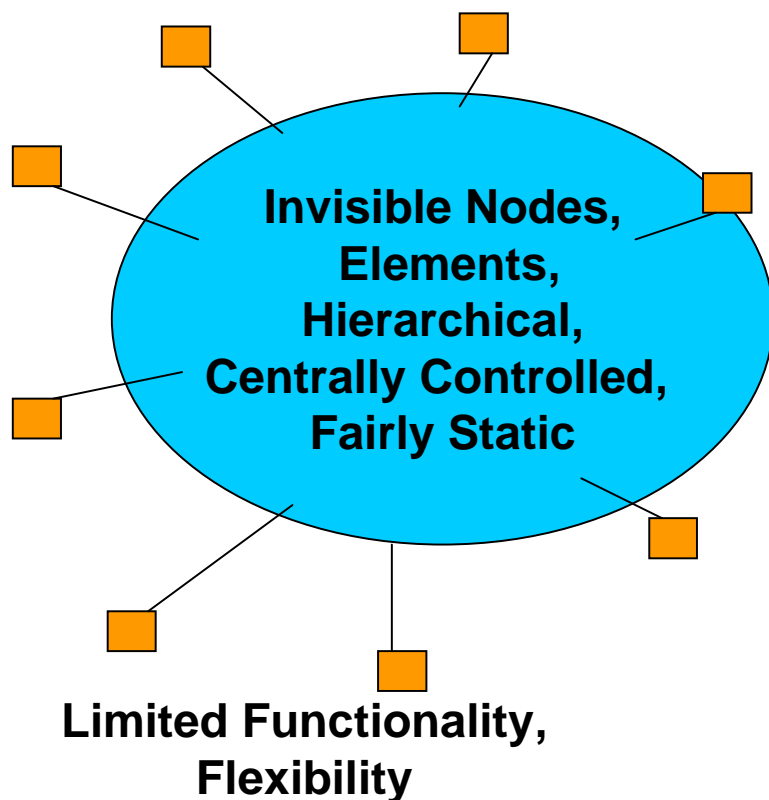




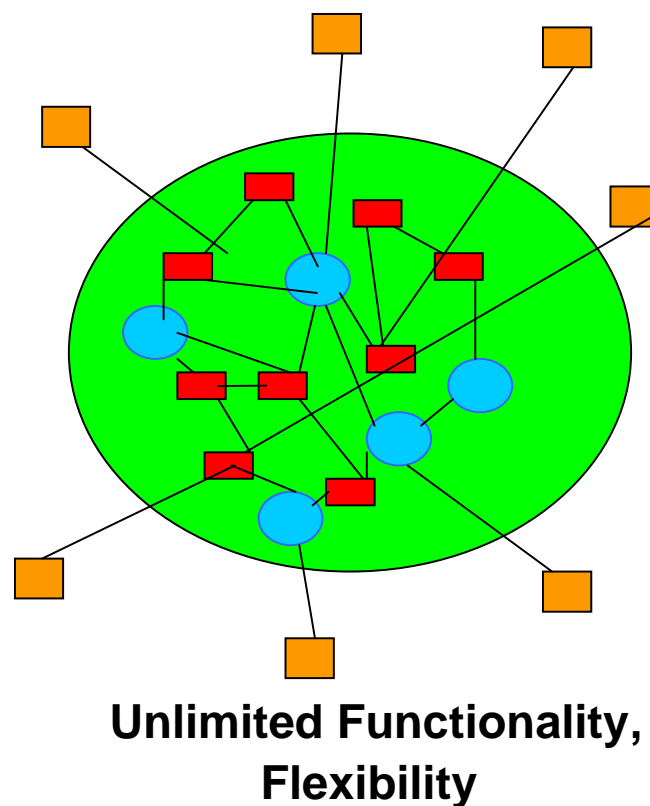
# Distributed Control... A Paradigm Shift...



**Traditional CCS:**  
**Invisible, Static Resources,**  
**Centralized Management**



**DCS: Dynamic Services,**  
**Visible & Accessible Resources,**  
**Integrated As Required By Apps**



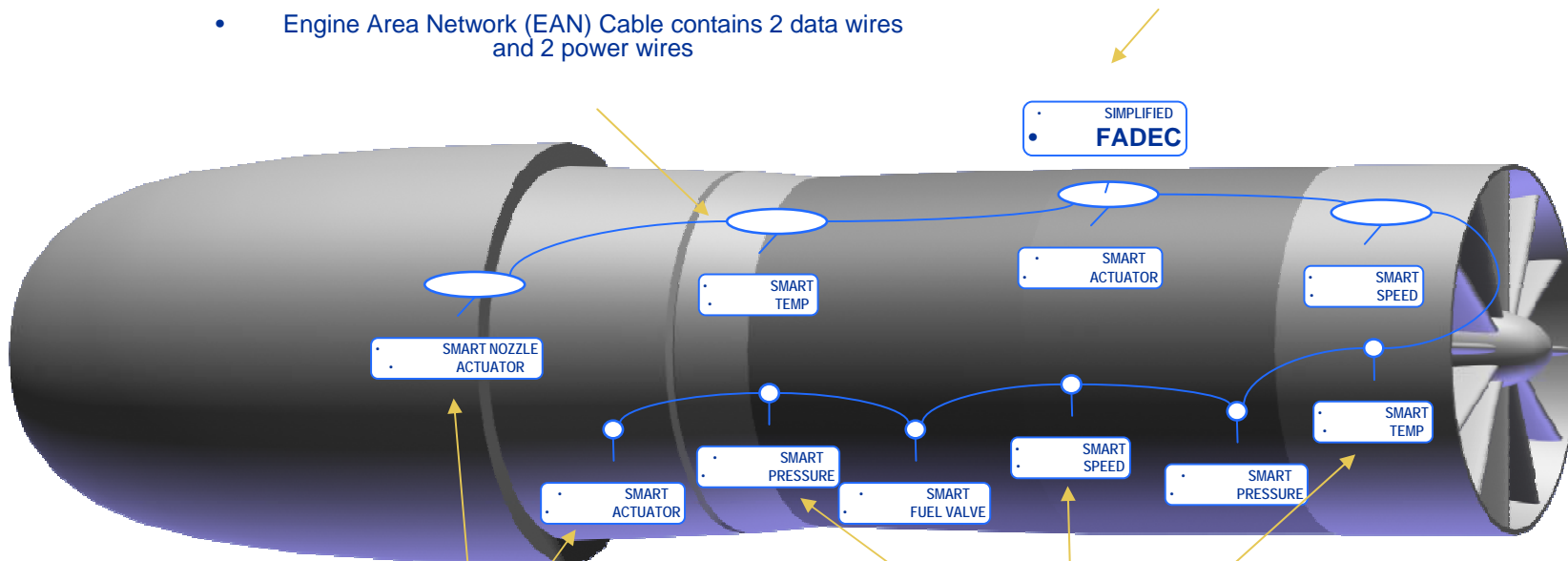


# High Temperature Distributed Control Systems A Historical Perspective ...“HiTeC”



- FADEC has conventional I/O removed
- EAN Interface and Distributed Power Supply added

- Engine Area Network (EAN) Cable contains 2 data wires and 2 power wires



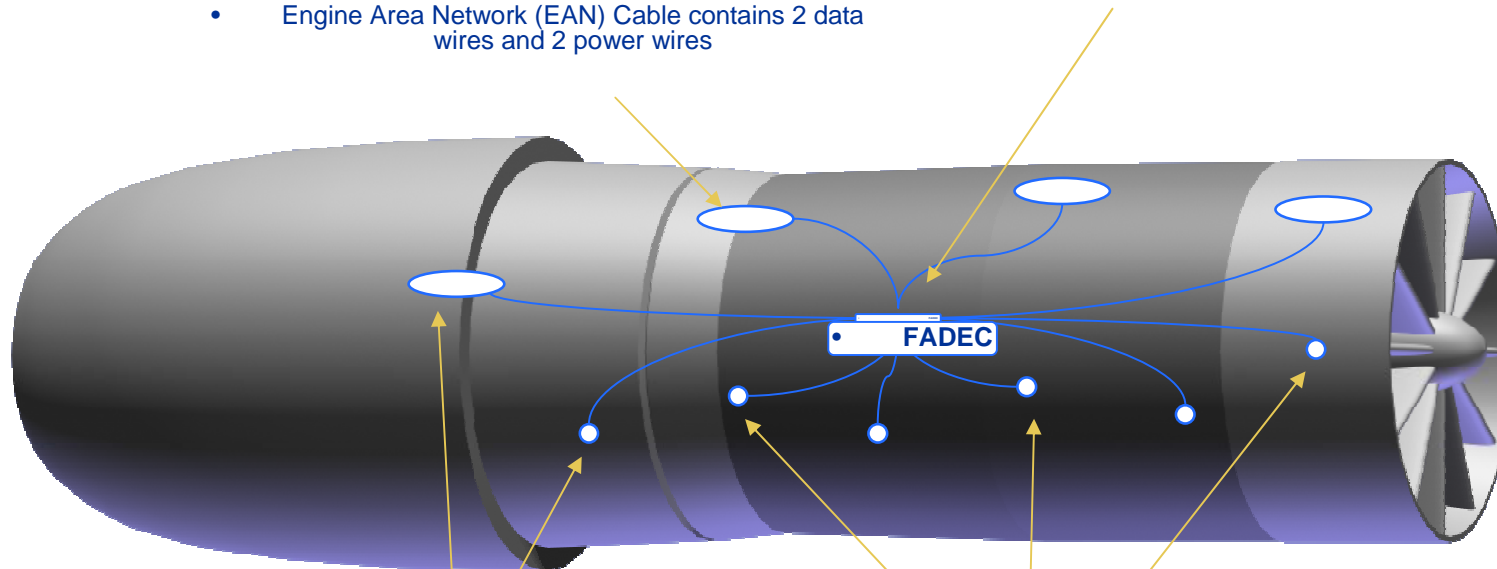
- Smart actuators close position loops locally
- Smart sensors provide outputs in engineering units (\*F, Hz, PSIA etc.)
- Smart devices (sensors or actuators) consist of baseline device plus Electronic Interface Unit (EIU)



# Current Centralized control System for Turbine Engine



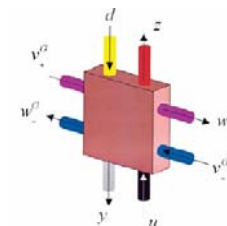
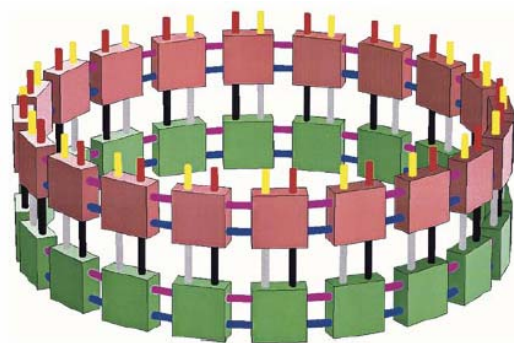
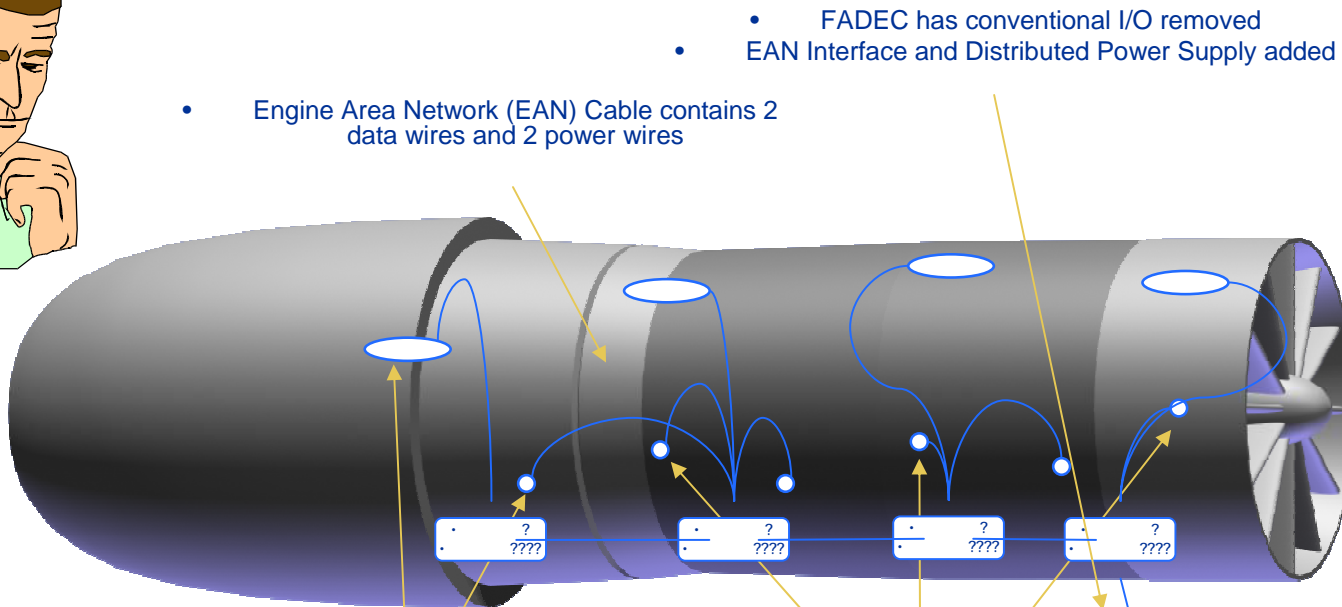
- FADEC has conventional I/O removed
- EAN Interface and Distributed Power Supply added
- Engine Area Network (EAN) Cable contains 2 data wires and 2 power wires



- Smart actuators close position loops locally
- Smart sensors provide outputs in engineering units (\*F, Hz, PSIA etc.)
- Smart devices (sensors or actuators) consist of baseline device plus Electronic Interface Unit (EIU)



# A “new” Distributed Control Systems for Engine Control

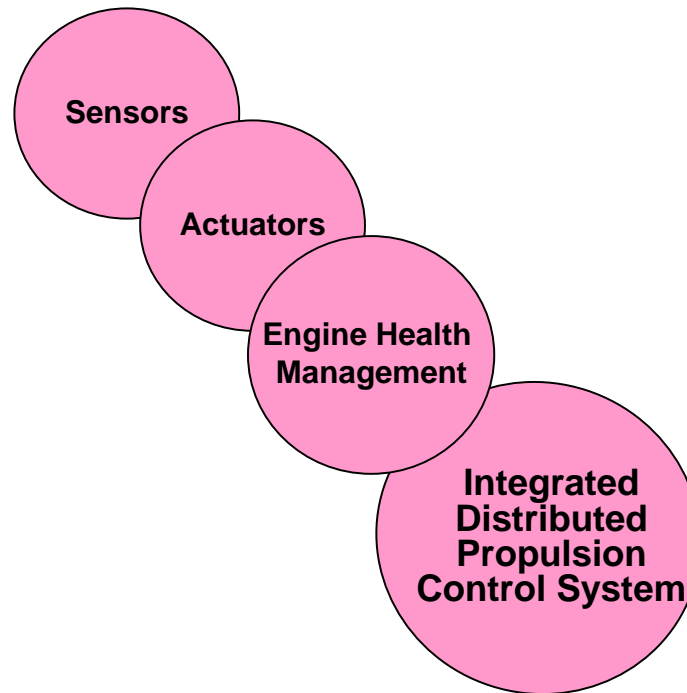




# Distributed vs. centralized tradeoffs



- List of common wins, loses, and draws
- In industry, tradeoff studies often try to justify things based on the “draws” instead of the “wins”



**changes must be considered in overall system context-**



# Comparison



**Table 1: Characteristic comparison of CCS and DCS.**

Hardware / Software Feature Comparison:	CCS	DCS
<i>controller or manager or processor</i>	centralized	distributed
<i>number of controllers or processors</i>	1	>1
<i>impact of obsolescence</i>	major	minimal
<i>status of links / network failures at any time</i>	✓	
<i>fault isolation with 100% certainty</i>	no	yes
<i>in-service reliability</i>		high
<i>cost of maintenance</i>	high	low
<i>harness weight / complexity</i>	high	low
<i>ease of adding system functionality</i>	hard	easy
<i>Ease of centrally located repair of one component</i>	✓	
<i>Device location plan and design</i>	✓	
<i>Accuracy of status of each component</i>		✓
<i>Impact of processor failure on overall system</i>	major	
<i>Ease of design of control algorithms</i>		✓
<i>quantity of CPUs</i>	2 with redundancy	large
<i>Each processor communicate with the master CPU</i>		✓
<i>Computing capacity of Central CPU</i>	very high	moderate
<i>reusability and maintenance of the software</i>	✓	✓✓✓
<i>Synchronization</i>	✓	✓
<i>Ease of Replacement and testing of individual modules</i>		✓
<i>Capabilities and user interfaces</i>		
<i>Scalability.</i>		high



## Premise of DCS



- The principles for designing DCSs are evolving
  - ✓ Industry is able to build DCSs, but is still learning how to engineer their design
- Engineering the architecture of DCSs will
  - ✓ Provide a basis for preventing the cost, schedule, and performance problems often experienced during development
  - ✓ Enable optimizing the architecture of the control system with respect to acquisition program criteria
  - ✓ Enable effective human-systems integration

**A methodology for engineering the architecture of DCS advances the state-of-the-art.**





# Centralized System Advantages



- **Simple programming model**
  - ✓ Ability to think about distributed architectures is an uncommon skill
- **Powerful CPU(s) needed**
  - ✓ Can use CPU for any needed function
  - ✓ Can adapt CPU loading to operating mode
- **Better operating environment for digital electronics**
  - ✓ Put machine in sheltered area away from combustion, environment
- **Arguably simpler software configuration**
  - ✓ All changes are made in one place in the system
- **Can grow up to limits of equipment rack**
  - ✓ More restrictive than one might think in a harsh environment system
- **Any of these reasons might be sufficient to justify a centralized system**

**“Put all your eggs in one basket and  
– watch that basket!” -- Mark Twain**



## When Is It “A Wash?” (no advantage)



- **Total system cost/weight**
  - ✓ Housing + cooling costs may outweigh wiring savings
  - ✓ DCS has components in harsher environment than CCS
- **System expandability**
  - ✓ Central system has limit on I/O connectors
  - ✓ DCS has limit on bus fanout
  - ✓ DCS has limited communication bandwidth
- **Inventory costs**
  - ✓ DCS have cheaper components, potentially but more kinds of them



## DCS architectures bring significant potential benefits



- ✓ Modularity – system integration of subsystems
- ✓ Flexibility – can scale up system and use heterogeneous components
- ✓ Diagnosability – can isolate faults more effectively
- ✓ Robust data transmission – network enables error control coding
- ✓ Flexible & incremental deployment – no single big box to buy up front
- ✓ Potentially improved fault tolerance & certifiability
- ✓ BUT, common purported advantages often don't materialize (cost, weight, expandability)
- **But, these benefits do not come for free**
  - ✓ All aspects of architecture must support distribution (software as well as hardware)
  - ✓ Distributed, concurrent design generally requires more sophisticated skills, tools, and infrastructure than centralized designs
- **Sometimes centralized is better**
  - ✓ Usually because it is easier to design/implement if you don't care about DCS advantages for a particular application

**Sometimes CCS is better than DCS...**



# DCS Tradeoff Pitfalls

- **DCS advantages are often subtle**
  - ✓ Require rethinking of system approach to be a win
  - ✓ Can appear as non-functional attributes: (diagnosability, maintainability)
  - ✓ May only be beneficial by making new functions easier to add: (flexibility)
- **DCSs also have scalability limits, but they are just different than for centralized systems**
  - ✓ Electrical fanout of buses / requires repeaters
  - ✓ Network bandwidth saturations / requires bridges and careful architecting
  - ✓ Complexity of distributed software (different than many are used to designing)
  - ✓ A poorly architected distributed system (especially just a porting of a centralized system) probably negates all benefits yet incurs extra cost for being
  - ✓ Distributed
- **DCSs require new skills**
  - ✓ Design & debug skills for concurrent, distributed systems
  - ✓ Maintenance and operation skills (e.g., network monitoring tools)



## DCS Tradeoffs...(Cont.)



- **Multiplexing control wires saves weight, wire cost, cable thickness**
  - ✓ One digital wire replaces multiple analog wires
  - ✓ Network must be fast enough to keep control loops closed
    - Much more on this later, but in general this can be done
    - Can use one wire per distribution node if network bandwidth is a concern
- **Network interface controller added to remote switching nodes**
  - ✓ Interfacing to even a simple network requires computer-like capability
    - In simplest case, computer just “muxes” wires
  - ✓ Local controller’s job is to translate control signals and switch power locally, with built-in health management
- **More complicated controllers permit functions to migrate**
  - ✓ Once we have a remote computers, why not do computation there beyond just network interface?
  - ✓ Carried to its logical conclusion, don’t even need the central computer anymore
    - But, doing this requires a significant rethinking of software architecture

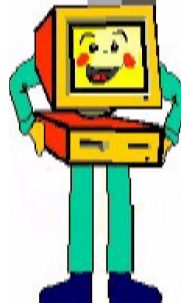
**This is not an easy solution...**



## Other Considerations...



- Model-Based Distributed Control...
- FADEC cooling
- Plug and Play Technology
- Actuation Technology
- Thermal Management
- Integrated Vehicle Health
- Subsystem/system Health: Diagnostics and Prognostics
- In-flight Support for Mission Effectiveness



Ground  
Analysis

On Board  
Health Management

- What are the missing gaps ?
- Visions of the new system
- The Challenge of the Changing System
- Traditional situation:
  - ✓ Future situation:
  - ✓ Model Based Verification and Validation of Distributed Control Architectures
  - ✓ Software / Hardware standardization will be realized in the future
  - ✓ Software as a contributor to cost/complexity
  - ✓ Controls will remain to be an important player in the propulsion system

### Future Health Management System Architecture

- Analysis Functions Integrated at Design
- Vehicle Onboard Health Designed for Flight, Checkout and Servicing
- High-Speed Processing for Real Time Health

**There are many other considerations when considering DCS...**



# DCS Advantages

## – Modularity, Flexibility, Adaptability, ...



- **Modular system development, support, and evolution**
  - ✓ A different team designing each node
  - ✓ Well-defined, tightly enforced interface (system message formats)
  - ✓ Can upgrade individual models and limit effect of changes on rest of system
- **Limits competition for resources among different features**
  - ✓ Can add computing I/O power incrementally
  - ✓ But, wastes resources on a node that might be inactive most of the time
    - Difficult to “time share” computing resources
- **Reduces interactions**
  - ✓ Easier to make worst-case guarantees on a per-module basis
  - ✓ Can re-certify only modules that have changed
  - ✓ Can have “critical” and “non-critical” modules, reducing certification effort

**DCS Advantages are yet to be proved for Propulsion System...**



# Higher Reliability for Propulsion Control



## Can We Get Extreme DCS?

- **What is the most extreme we can get with distributing functionality?**
  - ✓ How about one computer for every sensor and every actuator?  
MEMS technology potentially lets us do this
    - Sensors and actuators micro-machined from silicon
    - In some cases using CPU-capable process technology, so you can also get transistors on the very same piece of silicon

## Realities of Fine-Grain DCS

- **MEMS isn't quite here to this degree yet**
  - ✓ But, many systems use “smart sensors/actuators” that are a sensor or actuator paired with a CPU in a single package
- **Dividing up system this finely requires many changes**
  - ✓ Need a very decentralized, “fine grain” software architecture
  - ✓ Runs risk of saturating network network
    - Control loops can't be closed within regional CPUs, adding network traffic
    - Can address this with multiple bridged networks & clever architectural choices
- **Why we emphasize fine grain distribution**
  - ✓ Designing a system with this model in mind gives maximum flexibility
    - You can always aggregate software functions to co-exist in bigger nodes
    - But, tearing apart monolithic functions when you switch to smart nodes is very difficult

**“will it make a difference in engine maintenance?”**





## Another view of DCS vs CCS...



- **Centralized systems have one (or a few) controllers that coordinate all sensors and actuators**
  - ✓ All wire bundles lead to a single place
  - ✓ Power switching for loads is done at the centralized location
- **Distributed systems move functionality away from central point**
  - ✓ Distributed power switching replaces long power lines with control lines from central CPU
  - ✓ Networked control system muxes control signals to cut wire count from central CPU
  - ✓ Distributed control system puts compute power at remote power distribution
  - ✓ points to perform local control computations
  - ✓ Fine-grain distributed control system is an extreme – each sensor/actuator has its own computer
- **Hardware benefits are apparent – more flexibility; fewer wires**
  - ✓ But system-level win requires more subtle arguments

**Why Distributed Might Be Better?**  
**Lots of little things can be better than one big thing**  
**Extensibility / Flexibility / Task Partitioning**



# Why Are Distributed Systems Different?



- **Control flow must be decentralized**
  - ✓ Close fast control loops locally
  - ✓ Attempt to have weak dependence on other units for basic functionality (enables graceful degradation)
- **Data Flow is a limitation**
  - ✓ Latency – round trips on a network can cause control lag
  - ✓ Bandwidth – inexpensive networks have limited bandwidth
  - ✓ Reliability – networks drop packets due to noise, congestion, etc.
- **SW architecture should be compatible with HW architecture**
- Creating a good distributed architecture is more art than science

**Can't just chop up a centralized architecture and distribute it  
At least not if you want distributed advantages!**



## Summary Of Other DCS Advantages



- **Flexibility**
  - ✓ Can modify or upgrade systems by changing components
- **Robust data transmission**
  - ✓ Digital network lets you use error coding, controlling noise on signals
- **Simpler to build and maintain**
  - ✓ Single bus means you can't hook the wrong wires up – there is only one “wire”!
- **Enables fault tolerance**
  - ✓ A single CPU is a single point of failure
  - ✓ multiple CPUs support fault tolerance
- **Improves safety certifiability**
  - ✓ Separate CPU for critical functions means non-critical safety faults



Take a fresh look at DCS and see what it can offer to you...

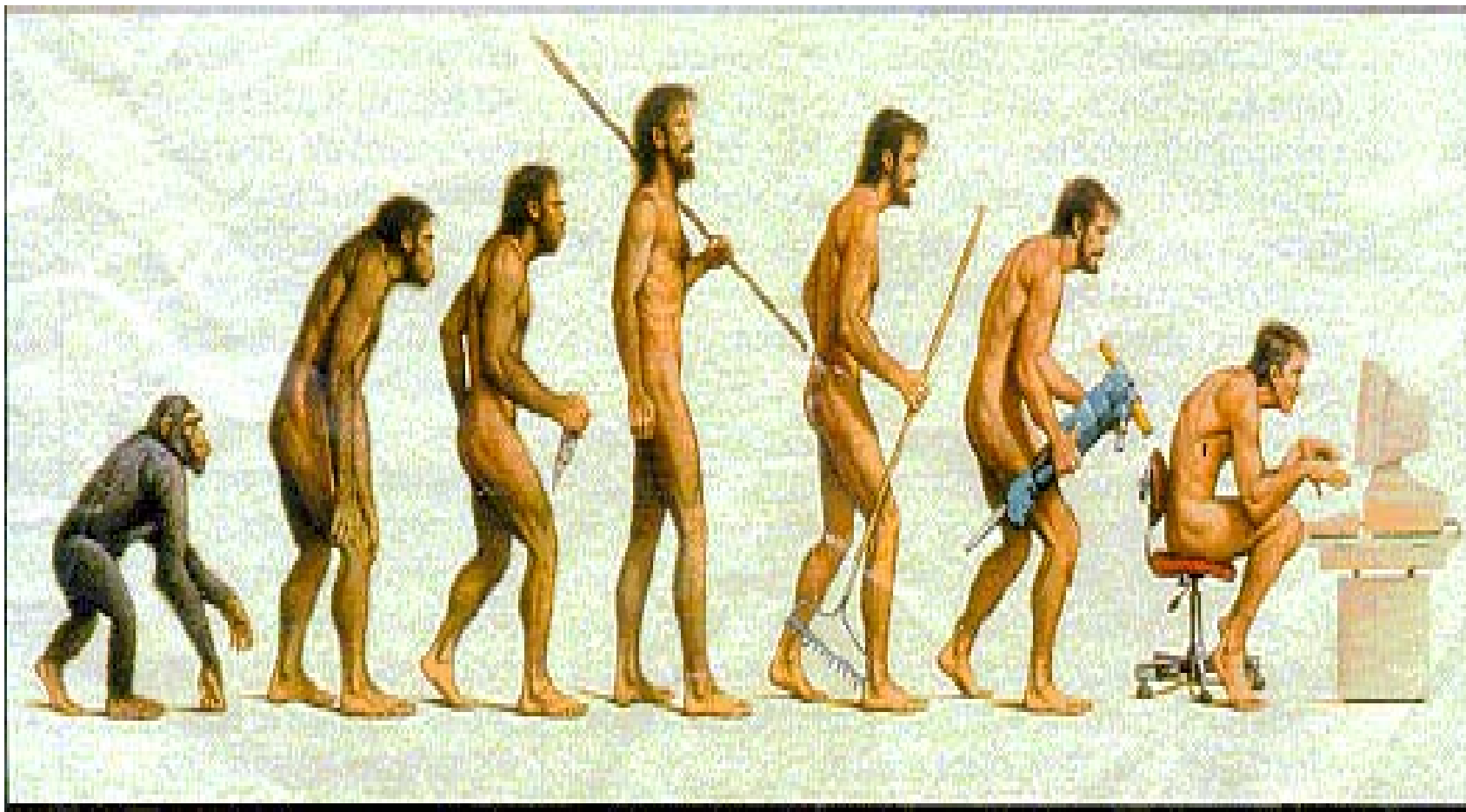


## Acknowledgment

- The author would like to thank **Dennis Culley** of the Advanced Controls Technology Branch at NASA Lewis Research Center and **Dr. Charles Cross**, Structures and Controls Branch Chief from Air Force Research Laboratory for their suggestions and comments regarding this paper.



# The Evolution Theory...



***Thank You!***



# BACKUPS



## Provides Essential Fault Tolerant Capability...



- **A single CPU has a single point of failure (the CPU)**
  - ✓ Duplicated hardware (multi-channel system) can help improve reliability
  - ✓ But a nasty fault or security breach can still slip through
  - ✓ And, a 2-of-2 system fails silent, does not fail operational
- **Distributed systems have greater fault tolerance *potential***
  - ✓ Different nodes can cross-check each other
  - ✓ Breaking into one node does not (necessarily) get you into other nodes  
If they don't have common mode software failures, system can be more robust
- **Distributed systems can tolerate arbitrary (uncorrelated) faults**
  - ✓ Multi-channel architecture without a central voter
  - ✓ Can fail operational by consensus voting to exclude faulty nodes from results



## More Flexible Deployment...



- **Buying a centralized computer can be a significant expense**
  - ✓ Creates significant barrier for someone on a budget (e.g., a homeowner)
  - ✓ Significant investment required before seeing any results at all
- **Sometimes phased deployment/upgrade is better**
  - ✓ Limited budget, want incremental improvements
  - ✓ Limited down-time for system during upgrades; need phased deployment
- **Distributed systems can help, if designed appropriately**
  - ✓ Replace old sensors/actuators with smart ones that are backward compatible
    - Any installed smart systems can provide incremental improvements
    - Can defer expense of central coordinating/optimizing compute nodes until sensors and actuators are in place for them to control
    - In the usual case incremental deployment has higher overall cost
      - But it is often the only practical way to accomplish business goals





## Simpler To Build And Maintain...



- **Single network wire vs. wiring harness**
  - ✓ Hard to connect to the wrong wire if there is only one wire
  - ✓ Thinner wire, lighter overall weight
  - ✓ Far fewer lightning protection devices (if applicable)
- **Maintain by replacing entire node**
  - ✓ Potentially easier on-line repair ("hot swap")
  - ✓ Can potentially function with one node broken or missing
- **Potentially takes no space at all**
  - ✓ Electronics can be stuffed into nooks and crannies of system
- **Potentially better error containment**
  - ✓ If one node fails, entire system does not fail...
  - ✓ as long as network does not fail and node did not have unique data





## Certiability...



- **Distributing functions potentially encapsulates changes**
  - ✓ Changing a non-critical node might not effect critical nodes
  - ✓ (But, be careful about indirect changes such as resource consumption)
- **Changing one critical node might not affect other critical nodes**
  - ✓ If system components can be certified individually
  - ✓ *AND*, each component depends only on advertised interfaces to other
  - ✓ components
  - ✓ *AND*, change on one component does not change interface
  - ✓ Then, *PERHAPS*, this means you don't have to recertify entire system
  - ✓ *BUT*, for now, this is a research hope more than a reality
  - ✓ It certainly is a way to reduce risk of certification problems by containing
  - ✓ changes even if you do have to recertify system just to be sure



## Flexibility...



- **Can add new components more easily**
  - ✓ Multiple vendors can add components to a well defined HW+SW standard interface
  - ✓ New components can have different physical size/shape as long as they can interface to the network
- **Scalable systems can be created on a pay-as-you-scale basis**
  - ✓ More copies of components added as system grows
    - (But, there are limits before repeaters are needed for network)
  - ✓ But, individual node packaging might be too much overhead if most systems have only 2 or 3 copies of a component
  - ✓ A single module with a couple long signal wires might be cheaper than a couple modules with a network wire